The Report Committee for Ingrid Ristroph
certifies that this is the approved version of the following report:

Analysis of Errors and Improvements in Numerical Approximations and Methods in Secondary Mathematics Curriculum

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

John Luecke

Mark Daniels

Analysis of Errors and Improvements in Numerical Approximations and Methods in Secondary Mathematics Curriculum

by

Ingrid Ristroph, B.A.; B.S. Math.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Arts

The University of Texas at Austin

August 2013

Abstract

Analysis of Errors and Improvements in Numerical Approximations and Methods in Secondary Mathematics Curriculum

Ingrid Ristroph, MA

The University of Texas at Austin, 2013

Supervisor: John Luecke

This report discusses three topics relating to errors of numerical methods and to improvements of numerical approximations. The introduction connects these topics to the secondary mathematics curriculum. The three chapters which follow develop the three selected topics: improving approximations of irrational numbers, error analysis of numerical integration methods, and discretization versus rounding error in Euler's Method for solving ordinary differential equations. The conclusion describes specific national secondary mathematical standards and classroom activities relevant to numerical approximations and error analysis.

Table of Contents

List of Figures

Chapter 1: Introduction

The primary uses of computers in mathematics classrooms are: (1) direct delivery of content, (2) student-led exploration or experimentation (i.e., parameterized "sliders" on graphing tools, probability simulations, and other applets), and (3) assessment of student knowledge and skills. Ironically, computers are rarely utilized to compute or calculate in secondary mathematics. As ever-increasing funds and discussions are committed to bringing computers into mathematics classrooms, educators need to be able to determine how technology can be used most effectively to improve student learning of mathematics. National standards for secondary mathematics call for the use of technology to find approximations for π, e, derivatives, integrals, and roots of real-valued functions. However, there is little acknowledgement of how these numerical approximations are made and to an even lesser degree what the associated errors are, or how to improve the error of these approximations.

The focus of this report is a brief summary of some numerical approximations and methods related to topics in secondary mathematics, ways to improve their accuracy, and an analysis of the errors of these approximations. The discussion here includes approximation of irrational numbers with series, error analysis for numerical integration methods, and Euler's Method for solving ordinary differential equations. Specifically, the speed and efficiency of convergence of a series, the precision, error bounds, and the distinctions between round off error and discretization error are considered.

Increasingly sophisticated computer graphics are helpful in visualizing functions and assessing the accuracy of estimations. Students of calculus learn the various types of Riemann Sums (left, right, Midpoint, Trapezoid) and Simpson's Rule while learning how to mathematically integrate. In these studies, students explore beautiful geometric

visualizations and move from discrete to continuous calculations of the area under a curve. High school students study errors generated by these methods graphically, as evidenced and assessed in Figure 1, a released College Board AB/BC 2003 exam question.

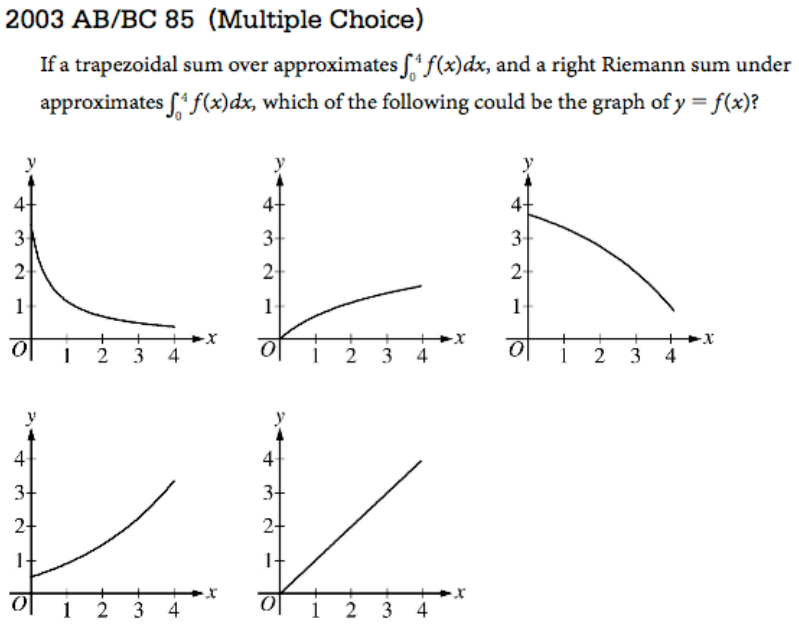

Figure 1. Item concerning error bound in numerical integration from Calculus AB/BC Exam [**1,** p. 12]

With the aid of graphics, students can improve their understanding of error bounds and develop an understanding of the magnitude of the errors arising in numerical integration approximations.

Fostering students' skills in numerical methods provides opportunities for students to deepen their conceptual understanding of fundamental ideas that improve their ability to solve problems in scientific and mathematical fields. For example, students are introduced to irrational numbers in middle school by estimating the value of

π by calculating the ratios of the circumference to the diameter of several circles. Similarly, computer scientists use the accuracy and speed of algorithms that approximate π to gauge the computing power of supercomputers. [7]

As computers become more common in mathematics classrooms, it is important that students not become passive users of such technology. Insights into the programming and calculations of numerical approximations and into the issues of efficiency and error will equip them well for an ever more technological future.

Chapter 2: Improving Approximations For Irrational Numbers

The logarithmic constant *e* is the limit of $\left(1+\frac{1}{n}\right)^n$ as *n* approaches infinity. The Maclaurin series expansion, or *the Direct Method* for approximation for *e* is

$$\sum_{k=0}^{\infty}\frac{1}{k!}=\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}+... \quad (1)$$

Efforts to calculate *e* to greater precision have largely been accomplished through improvement of computer techniques. The Direct Method itself can be rewritten as an equivalent series using *compression* so that it converges faster [**3**, p. 35]. Compression of a series is the algebraic combination and simplification of existing terms with a modification of the index of summation [**3**, p. 35]. A second technique of *powering* increases the rate of convergence for *e*. Powering is applied to an already compressed summation of e^x and uses small values of *x* and then exponentiates by $\frac{1}{x}$ to approximate *e* [**3**, p. 38]. Examples of both techniques will be shown after a discussion of how the speed of convergence is measured.

A comparison of the rates at which two series converge to the same limit commonly is done in two ways: decimal place accuracy and computation "run time costs" [**3**, p. 34]. The assessment based on decimals places is mostly algebraic. For example, one would compute the approximation with $k = 100$ terms in the series for the two different series used to approximate *e* and using the decimal place accuracy, note

which series is converging faster. The second way of comparing two series utilizes computational time. When the two series are computed on equivalent machine setups (i.e. same processor performance, memory capacity, software, etc.), the algorithm that requires fewer or simpler processor operations is said to converge at the faster rate.

A more efficient algorithm can be written using *pairwise series compressions*. The concept of series compression is simple. It is an algebraic combination of existing terms to the familiar series approximating *e*:

$$\frac{1}{n!}+\frac{1}{(n+1)!}=\frac{1}{(n-1)!n}+\frac{1}{(n-1)!n(n+1)}=$$

$$\frac{1(n+1)}{(n-1)!n(n+1)}+\frac{1}{(n-1)!n(n+1)}=\frac{(n+1)+1}{(n-1)!n(n+1)}=\frac{n+2}{(n+1)!}.$$

So the summation can be rewritten by replacing the *n* with 2*k* so that the index of summation reflects the combination of the two terms:

$$e=\sum_{k=0}^{\infty}\frac{1}{k!}=\sum_{k=0}^{\infty}\frac{2k+2}{(2k+1)!}. \tag{2}$$

For k = 19, this new series is accurate to 47 decimal places, while The Direct Method is accurate to 18 decimal places. When approximating *e* to 200,000 decimal places either one of these compressed summations are more than twice as fast as the Direct Method [**3**, p. 36].

There are two other ways to rewrite more efficient summations to approximate *e*: (1) compression of e^x and then exponentiating by $\frac{1}{x}$ to approximate *e*, and (2)

compressions composed of more than pairwise combination of terms. Explanations follow.

After the power series for e^x has been made more efficient using the technique of compression, the second technique of powering is applied to the power series for e^x to estimate *e*. Choose *x* so it is small and in some form of 2^{-n}. This way the result is squared *n* number of times to approximate *e*. Begin by using the general series expansion for e^x:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = \frac{1}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + ..., x \in \mathrm{R}. \tag{3}$$

The power series in (3) is then compressed using pairwise compression and then the series is evaluated for $x = \frac{1}{2}$ in the compressed series:

$$e^{\frac{1}{2}} = \sqrt{e} = \sum_{k=0}^{\infty} \frac{4k+3}{2^{2k+1}(2k+1)!}. \tag{4}$$

Squaring the result of (4) gives *e* accurate to 59 decimal places [**3**, p. 38]. The convergence of these power series improves dramatically for smaller *x* because the speed of convergence of a series is determined by the rate at which the denominators increase relative to the numerators. For example, for $x = \frac{1}{16}$ and the same number of *k* terms, square that approximation four times, yields *e* accurate to the 94 decimal places [**3**, p. 38]. So by using the technique of *powering* it is possible to achieve rapid convergence. The techniques for approximating *e* presented here are only faster by about 1.5% than other notable methods for speeding up evaluations of The Direct Method, such as the

Binary Splitting Method. But the methods discussed here are simple enough to explore their application to other series, such as those that approximate π, and may generate interest in more advanced techniques [**3**, p. 39].

Chapter 3: A Visualization of Error Bounds for Numerical Integration

In this chapter, pictorial analyses of error bounds are shown to lead to an improvement of a familiar elementary method for numerical integration. Riemann's sums and Simpson's methods are instances of Newton-Cotes quadrature that numerically integrates a function using equally spaced nodes, $x_0 < x_1 < \ldots < x_k$ in the interval $[a, b]$. These methods all use a sum, $\sum_{i=0}^{k} \Delta x_i \cdot f(x_i)$, to approximate $\int_a^b f(x)\,dx$. Newton-Cotes methods and their associated errors will be considered first. Then an improvement to one of these methods will be shown to lead to a more sophisticated method in which the points for evaluation of the integrand are chosen in an optimal way, rather than simply equally spaced.

The error for each method is defined to be the amount that needs to be added to the approximation to make it exactly equal to the definite integral.

$$\int_a^b f(x)\,dx = \text{ approximation} + \text{error}$$

Numerical integration approximations are often needed when the definite integral cannot be expressed in terms of elementary functions capable of evaluation; in such cases the exact error is often unknown. However, an upper and lower bound on the error may be feasible. In addition to providing a quantitative measure of the error bound, the bounds can be used to indicate the effect of the number of intervals of the evaluation, and an

illustration of the dependency of the error bound on the degree of smoothness of the integrand.

To understand the errors generated by these estimates, consider the simple cases of the left point and right point rules for computing Riemann sums. Both methods may either overestimate or underestimate the definite integral of a polynomial of a degree greater than degree zero.

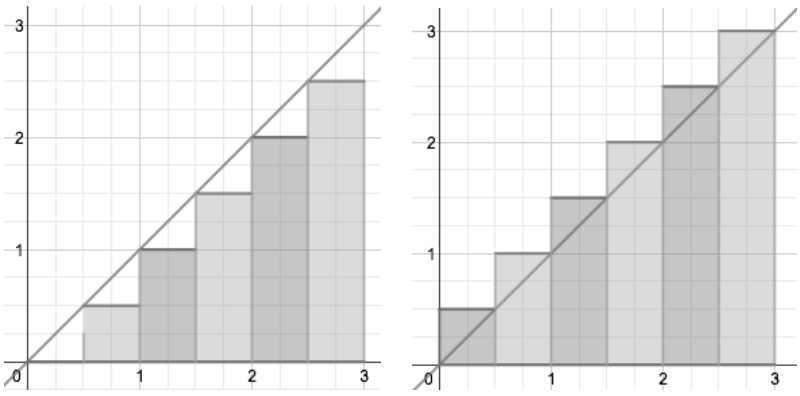

Figure 2. Left and right Riemann sums for monotonically increasing $f(x)$.

As illustrated in Figure 2, for monotonically increasing functions, the left point Riemann sum underestimates and the right point Riemann sum overestimates the integral. For such functions, the integral is bounded as follows:

$$\text{left Riemann sum} \le \int_a^b f(x)\,dx \le \text{right Riemann sum}$$

In a similar manner, as illustrated in Figure 3, for a monotonically decreasing function the left endpoint rule overestimates and the right endpoint rule underestimates the integral.

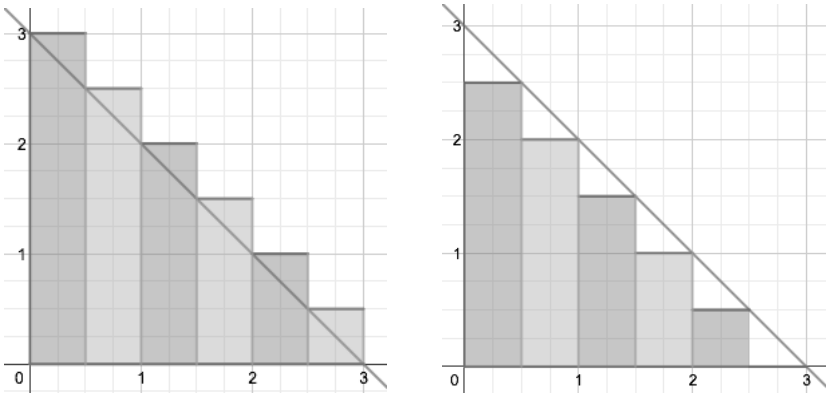

Figure 3. Left and right Riemann sums for monotonically decreasing $f(x)$.

For such functions, the integral is bounded as follows:

$$\text{right Riemann sum} \le \int_a^b f(x)\,dx \le \text{left Riemann sum}.$$

So, for either the left or right endpoint rule, the error bound is:

$$\left|\text{Error}\right| \le \left|\text{right Riemann sum} - \text{left Riemann sum}\right|.$$

The left and right Riemann's sum methods will give an exact approximation for polynomials of order 0. However, when the order of $f(x)$ is increased, errors result. The error is bounded as $\left|\int_a^b f(x)\,dx - \text{Left or right estimation}\right| \le \frac{M(b-a)^2}{2n}$ where M is the largest value of $\left|f'(x)\right|$ on the interval [a, b] [**1**, 40]. The error is bounded by the constant

$\frac{M(b-a)^2}{2}$ multiplied by $\frac{1}{n}$. Doubling the number of intervals will decrease the error bound by a factor of $\frac{1}{2}$. More importantly, because *M* is the largest value of $f'(x)$ on [*a*, *b*], the error depends on how steeply the graph is rising or falling (or how rapidly the rate of change of a quantity is varying). Figures 4 and 5 illustrate this.

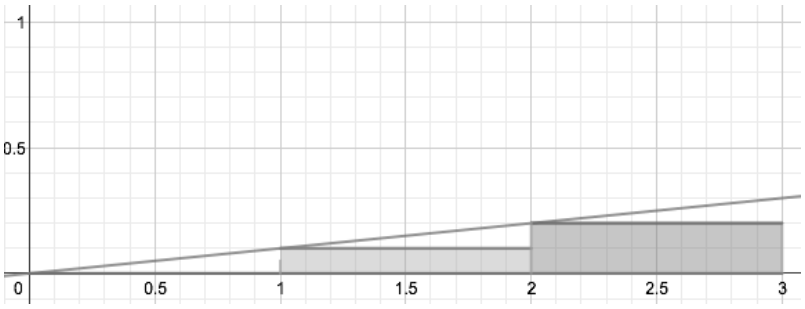

Figure 4. Integral of $y = 0.1x$ using left Riemann sum.

Figure 4 shows the approximated integral of $y = 0.1x$ from $x = 0$ to $x = 3$ using three intervals of left Riemann sums. This area estimation is 0.3, while the actual area is $\int_0^3 0.1x\ dx = 0.45$. The error for this approximation is 0.15.

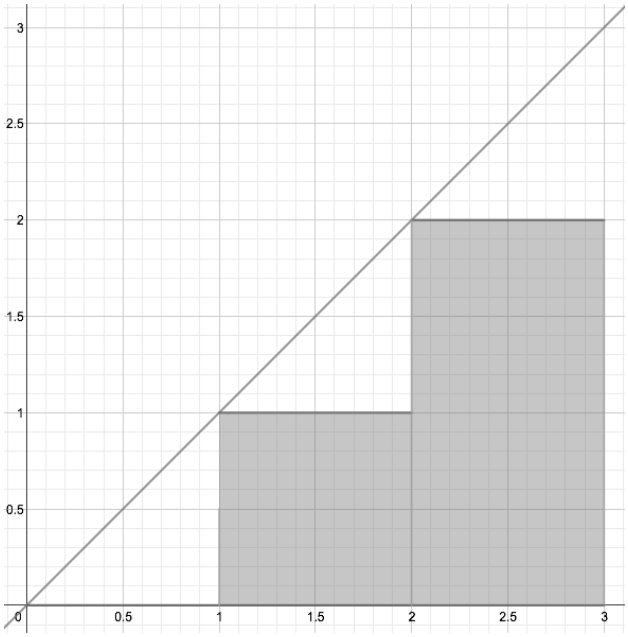

Figure 5. Integral of $y = x$ using left Riemann sum.

Figure 5 shows the approximated integral of $y = x$ from $x = 0$ to $x = 3$ using three intervals of left Riemann sums. This area estimation is 3, when the actual area is $\int_0^3 x\,dx = \frac{9}{2} = 4.5$. The error for this approximation is 1.5. These examples also demonstrate that the linear function with the greater value of $f'(x)$ also has the greater error.

The approximations from the Trapezoid and Midpoint Rules are more accurate than either the right or left endpoint rules. For what characteristics of $f(x)$ does the Trapezoid Rule over and underestimate?

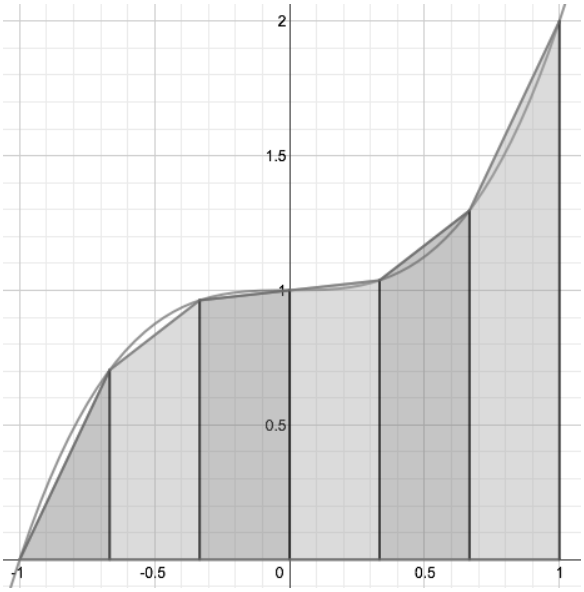

Figure 6. Trapezoid Rule for a cubic polynomial.

The graph of the cubic polynomial in Figure 6 illustrates an interval, from -1 to 0, for which the Trapezoid Rule underestimates and another interval, from 0 to 1, for which the Trapezoid Rule overestimates. If the function is concave downward, or $f''(x) < 0$, on the interval [a, b], then the secant lines forming the tops of the trapezoids are below the function; so the Trapezoid Rule underestimates. If the function is concave up, or $f''(x) > 0$, on the interval [a, b] then the secant lines forming the top of the trapezoids are above the function and the Trapezoid Rule overestimates.

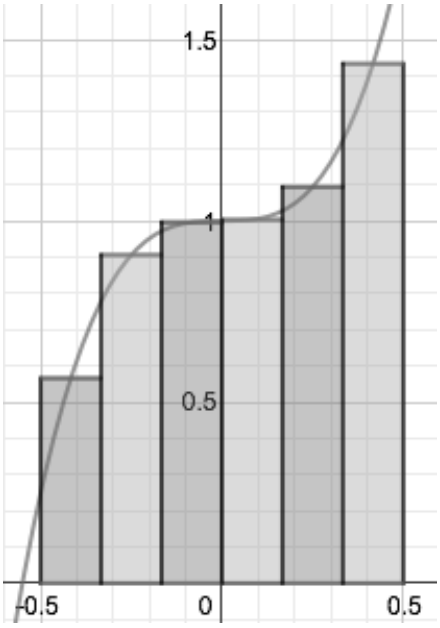

Figure 7. Midpoint Rule for a cubic polynomial.

Conversely, as observed in Figure 7, for the interval [-0.5, 0] in which $f(x)$ is concave down, the midpoint rule overestimates. Likewise, for the interval [0, 0.5] in which $f(x)$ is concave up, the midpoint underestimates.

The Midpoint Rule typically will be more accurate than the Trapezoid Rule as suggested by Figure 8. The area of the rectangle in the Midpoint Rule is the same as the area of trapezoid *ABCD* on the left side of Figure 8. Using this quadrilateral instead of the rectangle, the two trapezoids can be compared in the right of the figure. From Figure 8, we can see that midpoint error (area denoted with slashed lines) is less than the trapezoid error (area denoted by shading). [**6**, p. 460].

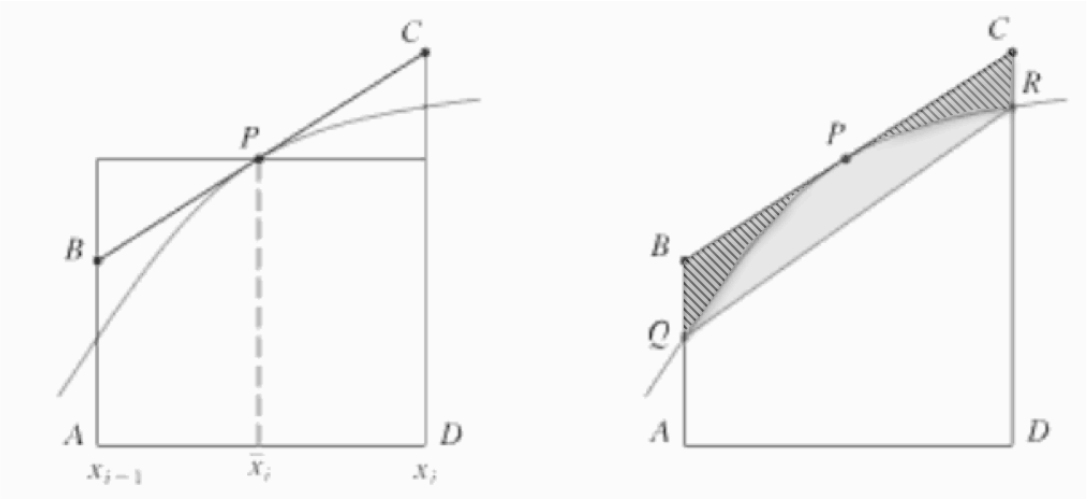

Figure 8. Comparison of Midpoint and Trapezoid Rule error. [**6**, p. 460][1]

The error bounds for Midpoint and Trapezoid rule are both dependent upon the degree of concavity of $f(x)$. If $|f''(x)| \le K$ for $a \le x \le b$ and the number of subintervals equals n, then $|\text{Error of Trapezoid}| \le \frac{K(b-a)^3}{12n^2}$ and $|\text{Error of Midpoint}| \le \frac{K(b-a)^3}{24n^2}$. [**6**, p.460] By comparison of the denominators of the error bounds, suggests that the size of the error in the Midpoint Rule is about half the size of the error of Trapezoid Rule.

The following is an improvement of the Trapezoid Rule; this improvement of the method exactly integrates polynomials through degree three. [**4**, p. 48]. Consider the parabola $y = x^2$ on the interval [-1, 1] in Figure 9. If the trapezoid method were to be used to approximate the integral on the [-1, 1], the approximating trapezoid is a rectangle with vertices (-1, 1), (1, 1), (1, 0) and (-1, 0) and area = 2. This approximation is far larger than the actual area, $\int_{-1}^{1} x^2 dx = \frac{2}{3}$. If the approximating trapezoid had less height, then the

[1] This image has been modified from the original to include the shading of regions.

approximation could be improved. If the height is brought down, closer to the *x*-axis in this case, where the two shaded areas cancel out in Figure 9, an exact area could be found.

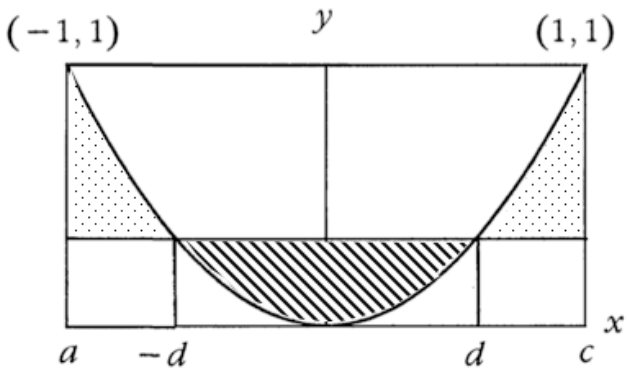

Figure 9. Parabola evaluated using improved trapezoid method [**4**, p. 47] [2]

The parabola intersects the top of the trapezoid at the point (d, d^2). The height of the approximating area is d^2 and the width is 2, yielding an area of $2d^2$. When does the cancelation of areas occur? To find the points, d and $-d$, at which this approximation equals the exact area: $2d^2 = \frac{2}{3}$, $d^2 = \frac{1}{3}$, $d = \pm\frac{1}{\sqrt{3}}$.

Interestingly, this procedure works exactly and generally for quadratics. The exact area under $f(x) = Ax^2 + Bx + C$ from $x = a$ to $x = c$ is the same as a trapezoid with base width $(c - a)$ and whose height is determined by the points along the parabola that are pulled in by a factor of $\frac{1}{\sqrt{3}}$ towards the center. The base of the trapezoid is

[2] This image has been modified from the original to include the shading of regions.

$\left(\frac{a+c}{2}\right)\pm\frac{1}{\sqrt{3}}\left(\frac{a-c}{2}\right)$ rather than previously $\left(\frac{a+c}{2}\right)\pm 1\left(\frac{a-c}{2}\right)$, or *a* and *c*. This can be shown by directly integrating the standard quadratic function, $f(x)=Ax^2+Bx+C$.

$$\int_a^c (Ax^2+Bx+C)dx=\frac{A}{3}(c^3-a^3)+\frac{B}{2}(c^2-a^2)+C(c-a) \tag{5}$$

The base of the trapezoid ($c - a$) will factor into this area (5) the following number of times:

$\frac{A}{3}(a^2+ac+c^2)+\frac{B}{2}(a+c)+C$. Algebraically we can simplify this previous expression and show that it is equivalent to the average height of the trapezoid of the improved method:

$$\frac{1}{2}\left[f\left(\frac{a+c}{2}+\frac{1}{\sqrt{3}}\left(\frac{a-c}{2}\right)\right)+f\left(\frac{a+c}{2}-\frac{1}{\sqrt{3}}\left(\frac{a-c}{2}\right)\right)\right].$$

The integral, or the area of the trapezoid, is

$$\frac{1}{2}\left[f\left(\frac{a+c}{2}+\frac{1}{\sqrt{3}}\left(\frac{a-c}{2}\right)\right)+f\left(\frac{a+c}{2}-\frac{1}{\sqrt{3}}\left(\frac{a-c}{2}\right)\right)\right]\bullet\left(c-a\right).$$

For a domain discretized into *k* equally spaced intervals about $k+1$ points, the integral is equivalent to the improved Trapezoid rule is

$$\int_a^c f(x)\,dx\approx\frac{1}{2}\sum_{i=0}^{k}\left[f\left(\frac{x_i+x_{i+1}}{2}+\frac{1}{\sqrt{3}}\left(\frac{x_i-x_{i+1}}{2}\right)\right)+f\left(\frac{x_i+x_{i+1}}{2}-\frac{1}{\sqrt{3}}\left(\frac{x_i-x_{i+1}}{2}\right)\right)\right]\bullet\left(x_{i+1}-x_i\right).$$

The improved trapezoid method integrates quadratics exactly because each subinterval is the same as the exact integral. Surprisingly, the new method also integrates cubic functions exactly and can be shown using the same rationale [**4**, p. 48].

This approach of symmetrically moving the altitude base points can be used to improve Simpson's method so it evaluates polynomials through degree four and five exactly. Both the improved trapezoid and Simpson methods are cases $n = 2$ and 3 of Gaussian *n*-point quadrature. [**4**, p. 50]

However, it is worth pointing out that this improvement relies on knowing values of the function at points that are different from the original discrete points or nodes, x_i. So, the improved method requires additional evaluations of the function. This limits how the method could be used. For example, if the function is measured experimentally at discrete locations x_i, we do not have access to the values of *f* between x_i and x_{i+1}, since we didn't measure there.

Chapter 4: Discretization vs. Rounding Error in Euler's Method

Numerical methods are often used to approximate solutions of differential equations when analytical techniques such as direct integration or series expansions do not apply or are not feasible. The simplest and oldest approach is called Euler's method. There are two versions of Euler's method; they demonstrate the tradeoff between *discretization error* and *rounding error* [**2**, p. 396]. Discretization error results from numerical methods that use finitely many known parameters to approximate the exact solution of a differential equation. Rounding error results when a stored number, due to a computer's limitations, differs from its true value. Rounding errors are propagated in iterative algorithms when rounded values are transported from one step to the next in the approximation.

Consider the differential equation $\dot{x} = f(x,t)$ with the initial condition $x(0) = x_0$. As shown in Figure 10, the horizontal axis, t, is discretized and divided into equally spaced intervals using a uniform stepsize, $h = T/n$ for a positive integer n. Euler's method generates a broken line approximation to the exact solution at various lattice points, $t_k = kh$ for $k = 0, 1, 2, \ldots n$, on the interval $[0, T]$.

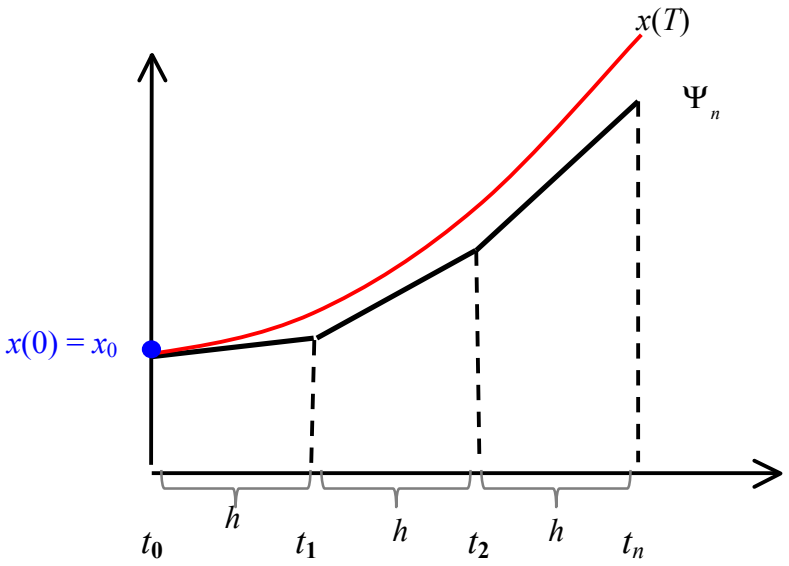

Figure 10. Euler's Method for solving an ODE

Euler's solution, Ψ_n, is recursively defined (5) using the previous point's vertical height plus the product of the horizontal shift and the slope of the previous point. The method begins by setting $\Psi_0 = x_0$.

$$\Psi_{k+1} = \Psi_k + hf(\Psi_k, t_k) \text{ for } k = 0, 1, \ldots, n-1 \qquad (5)$$

A second version of Euler's method, called the *quadrature form*, is derived using the left Riemann sum as an approximation for the definite integral for the initial value problem.

$$x(T) = x_0 + \int_0^T f(x,t)\,dt$$

$$x(T) \approx x_0 + h\sum_{i=0}^{n-1} f(x(t_i), t_i)$$

Analogously, the quadrature form of Euler's Method (6) uses sums of slopes, S_n, to generate an approximate solution, Φ_n .

$$\Phi_n = x_0 + hs_n \tag{6}$$

The sum of the slopes, s_n, at the lattice points, t_i, with $s_1 = f(x_0, 0)$ and $k = 1, 2, \ldots, n-1$ are generated as follows:

$$s_{k+1} = s_k + f(x_0 + hs_k, t_k) .$$

In order to compare the two forms of Euler's method computationally, it is necessary to have agreed upon industry standards. There are guidelines called IEEE 754 set by the Institute of Electrical and Electronic Engineers that govern how numbers are represented, rounded, operated, and stored in computer memory. The actual computer output of this single-precision floating-point evaluation is denoted by *fl*(expression) [**2**, p. 397]. The numerical methods will be evaluated using this floating-point arithmetic denoted by $fl(\Psi_n)=\hat{\Psi}_n$ and $fl(\Phi_n)=\hat{\Phi}_n$ [**2**, p. 397].

Theoretically, reducing the width of the stepsize in either method decreases discretization error and results in a more accurate solution. However, it emerges that decreasing the stepsize in either version of Euler's method results in increasing the accuracy of the approximated solution only up to a certain point. An extremely small stepsize means that there is an increase in the number of steps required to cover the given interval [0, *T*]; with these additional iterations there are more computations work and a resulting accumulation of rounding error [**2**, p. 397].

A second cause for an increase in rounding error due to the structure of the standard Euler's method (5) will be examined by comparing the absolute errors of the two methods. The magnitude of the rounding error contributing to the absolute error can be reduced if, instead of using the standard Euler method, we use the quadrature form of Euler's method. To better understand the impact of this reorganized algorithm, we will look at two solutions using the quadrature and standard forms of Euler's method for successively smaller stepsizes. The first example is the exponential growth problem $\dot{x} = x$ on the interval [0,1] with the initial condition $x(0) = 9$. The exact solution is $x(t) = 9e^t$. Both methods are computed and the absolute errors, $x(T) - \hat{\Phi}_n$ and $x(T) - \hat{\Psi}_n$, are graphed below in Figure 11 [**2**, p. 398].

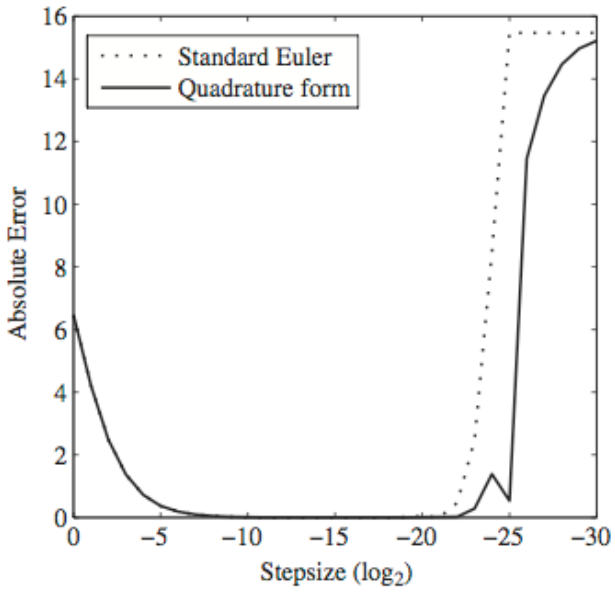

Figure 11. Comparison of absolute error in standard and quadrature Euler's Methods for the solution of $\dot{x} = x$ on [0, 1] with initial condition $x(0) = 9$. [**2**, p. 398]

There is an optimal range for stepsize (approximately from 2^{-10} to 2^{-22}) for both the standard Euler method and quadrature form. Initial decreases in the stepsize reduce the absolute error, but eventually the growth of the rounding error outweighs the benefits of the smaller stepsizes.

Consider the initial value problem $\dot{x} = -\sin(t)$ on the interval [0,1] with the initial condition $x(0) = 5000$. This second example better demonstrates the benefit of the quadrature form (6) because the slope field is dependent only upon *t*. The standard method computes $\Psi_n = x_0 - h\sin(t_0) - h\sin(t_1) - \ldots$ and the quadrature form computes $\Phi_n = x_0 - h\left(\sin(t_0) + \sin(t_1) + \ldots\right)$. Rounding errors are delayed with the quadrature form because it sums the slopes of all the lattice points before multiplying by *h*. Rounding error eventually impairs the standard method, Ψ_n, because our initial condition, 5000, is so large compared to the small values of $h\sin(t_k)$.

Figure 12 illustrates that if $h < 0.0003$ ($\log_2(0.0003) = -11.7$), then the error for the standard method is high. The *machine epsilon* is the smallest possible number that when added to 1 yields a result other than 1. The machine epsilon for IEEE 754 is $\varepsilon = 2^{-24}$ [**2**, p. 398]. Note that $5000\varepsilon = 0.0003$ [**2**, p. 398]. Thus, because *fl*(5000 + *h* sin(*t*)) = 5000, the algorithm for generating the solution line is unable to move away from the initial condition because the value for *h*sin(*t*) in floating-point evaluation is 0 for such small values of *h*. The quadrature method avoids this problem through a simple rearrangement. Although the two expressions of the Euler's method are algebraically

equivalent, its implementation uses ϕ_n which sums up the slopes, s_n, and then multiplies by h. Due to floating point evaluation's rounding errors, Φ_n is not computationally equivalent to Ψ_n.

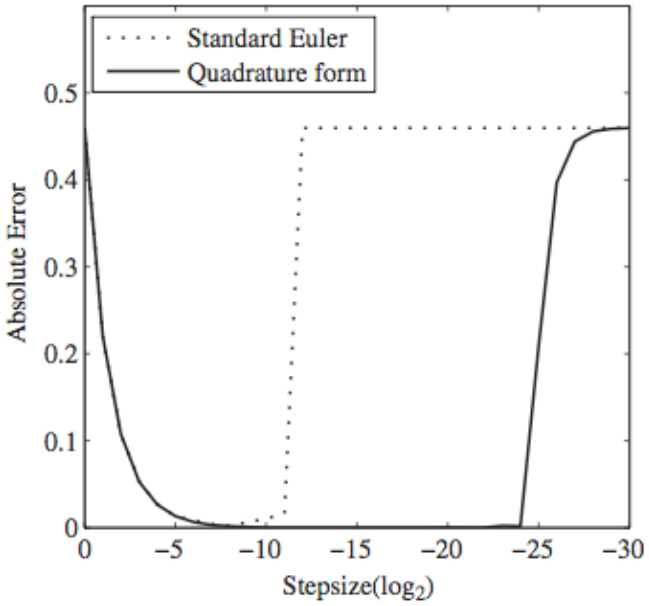

Figure 12. Comparison of absolute error in standard and quadrature Euler's Methods for the solution of $\dot{x} = -\sin(t)$ on [0, 1] with large initial condition [**2**, p. 399]

This study of errors from these two forms of Euler's method shows us three lessons:

1. Multiplying by very small h can cause numerical approximations to degrade. A wiser method is to sum slopes and then multiply that larger sum by h.
2. Although there is a theory which states that if IVP satisfies certain conditions then the absolute error of Euler's method will converge to 0 as h goes to 0 [**2**, p. 398], computer limitations may restrict practical application of the theory.

3. All good numerical algorithms need to be thoroughly vetted through error analysis. What may seem like a more elegant algorithm may not be the best computationally. Eliminating a step in the algorithm might actually cause absolute error growth in the long run.

Chapter 5: Conclusion

Secondary mathematics students are often awed by the power of technology. However, they should realize that there are limitations to computers and be able to "monitor and reflect on the process of mathematical problem solving" when using technology [**5**]. As processor speeds increase and memory capacity expands, numerical approximations continue to improve but will never be without error. High school mathematics students must "judge the meaning, utility, and reasonableness of the results of symbol manipulations, including those carried out by technology" [**5**]. There will always be finite resources (memory, processor speed) despite practical needs to compute approximations for infinite and infinitesimal concepts. This report connects with three concepts taught in a Calculus class: series to approximate irrational numbers, Riemann's sums to numerically integrate a function, and Euler's Method (part of the BC Calculus syllabus). Some specific ideas for middle and high school lessons exploring computational errors follow.

1. Computing and comparing the calculations and decimal place accuracy for 24π and $\underbrace{(\pi+\pi+\pi+....+\pi)}_{24 \text{ times}}$ on a computer algebra system such as Wolfram Alpha. (Although numerically equivalent, one crashes the website, the other doesn't!)
2. Approximating π^{10} using the formula for the sum of a geometric sequence and a rearranged formula.

3. Using a rearranged Quadratic Formula to avoid catastrophic cancellation when the quantities b^2 and $4ac$ are close in value.
4. Observing rounding error in compound interest problems.
5. Estimating decimal place accuracy and truncation of the infinite series for e and π and in other approximations of π such has $\frac{22}{7}$.
6. Studying the error in approximations of derivatives by difference quotients.
7. Exploring Newton's Method for finding zeros and recognizing cases for which the method fails to converge.

These explorations will offer opportunities for students to consider possible improvements in numerical approximations, to analyze various types of error, and to better appreciate the techniques arising in mathematical proofs- both in numerical analysis and in important theoretical investigations.

References

1. AP Calculus, College Board. 2008. Professional Development Workshop Materials. Retrieved April 12, 2013. http://apcentral.collegeboard.com/apc/public/repository/ap-sf-calculus-approximation.pdf

2. Borges, C. Discretization vs. Rounding Error in Euler's Method. *The College Mathematics Journal*. **42** (2011) No. 5 pp. 396–399

3. Brothers, H. (Jan., 2004). Improving the Convergence of Newton's Series Approximation for *e*. *The College Mathematics Journal*. Vol. 35, No. 1, pp. 34–39.

4. Kendig, Keith. (Jan., 1999). Pictures Suggest How to Improve Elementary Numerical Integration. *The College Mathematics Journal*. Vol. 30, No. 1, pp. 45–50.

5. Principles and Standards for School Mathematics. 2000. Retrieved November 17, 2012. http://standards.nctm.org/

6. Stewart, James. (1995). *Early Transcendentals 3rd Edition*. Brooks Cole, Pacific Grove.

7. U.S. Department of Energy. 2011. Supercomputers Crack Sixty-Trillionth Binary Digit of Pi-Squared. Retrieved April 12, 2013. http://energy.gov/articles/supercomputers-crack-sixty-trillionth-binary-digit-pi-squared

Vita

Ingrid Ristroph moved to Austin, Texas after graduating from Tomball High School in Tomball, Texas in 1999. She then earned her BS in Mathematics and a BA in Sociology from The University of Texas at Austin in 2003. After graduating, she authored and edited secondary mathematics instructional materials for Holt, Rinehart, and Winston and other publishers for two years. She then taught the following courses at Martin Middle School and Eastside Memorial High School in Austin, Texas for seven years: 8th grade math, Algebra I, Algebra II, Pre-calculus, and AP Statistics. In the summer of 2011, she entered the Graduate School at The University of Texas at Austin. She is currently is a Senior Program Coordinator at The Charles A. Dana Center at The University of Texas at Austin; there she works with mathematics teachers in large urban school districts to implement Texas and Common Core State Standards and to develop online courses (Grade 8 mathematics, Algebra II, Pre-calculus, Statistics, and a high school fourth year project-based course).

Email Address: ingrid.ristroph@austin.utexas.edu

This report was typed by the author.